



WHAT WE'RE DOING:

The final circuit is a bit of a test. We combine what we learned about using transistors in CIRC03 to control a relay. A relay is an electrically controlled mechanical switch. Inside the little plastic box is an electromagnet that, when energized, causes a switch to trip (often with a very satisfying clicking sound). You can buy relays that vary in size from a quarter of the size of the one in this kit up to as big as a fridge, each capable of switching a certain amount of current. They are immensely fun because there is an element of the physical to them. While all the silicon we've played with to this point is fun sometimes you may just want to wire up a hundred switches to control something magnificent. Relays give you the ability to dream it up then control it with your Arduino. Now to using today's technology to control the past. (The 1N4001 diode is acting as a flyback diode, for details on why it's there visit: <http://ardx.org/4001>)

THE CIRCUIT:

Parts:



**CIRC-11
Breadboard Sheet**
x1



**Diode
(1N4001)**
x1



**Transistor
P2N2222AG (TO92)**
x1



**Relay
(DPDT)**
x1



**2.2k Ohm Resistor
Red-Red-Red**
x1



**560 Ohm Resistor
Green-Blue-Brown**
x2



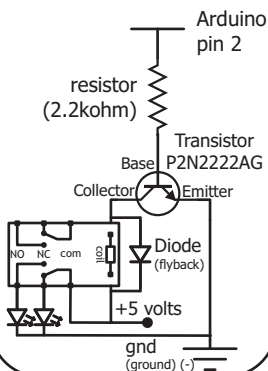
Green LED
x1



Red LED
x1

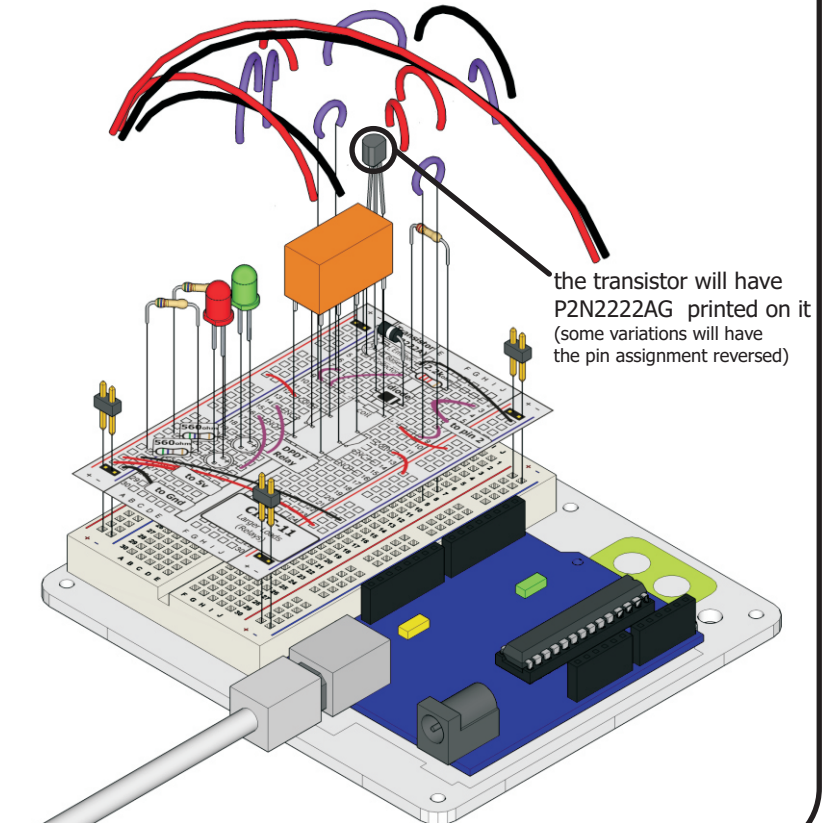
2 Pin Header
x4

Schematic



The Internet

..download..
breadboard layout sheet
<http://ardx.org/BBLS11>
..view..
assembly video
<http://ardx.org/VIDE11>



CODE (no need to type everything in just click)**File > Examples > 1.Basic > Blink**

(example from the great arduino.cc site, check it out for other great ideas)

```

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 2;                // ***** CHANGE TO PIN 2 *****

void setup()                   // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);     // sets the digital pin as output
}

void loop()                    // run over and over again
{
  digitalWrite(ledPin, HIGH);  // sets the LED on
  delay(1000);                 // waits for a second
  digitalWrite(ledPin, LOW);   // sets the LED off
  delay(1000);                 // waits for a second
}

```

NOT WORKING? (3 things to try)**Nothing Happens**

The example code uses pin 13 and we have the relay connected to pin 2. Make sure you made this change in the code.

No Clicking Sound

The transistor or coil portion of the circuit isn't quite working. Check the transistor is plugged in the right way.

Not Quite Working

The included relays are designed to be soldered rather than used in a breadboard. As such you may need to press it in to ensure it works (and it may pop out occasionally).

MAKING IT BETTER**Watch the Back-EMF Pulse**

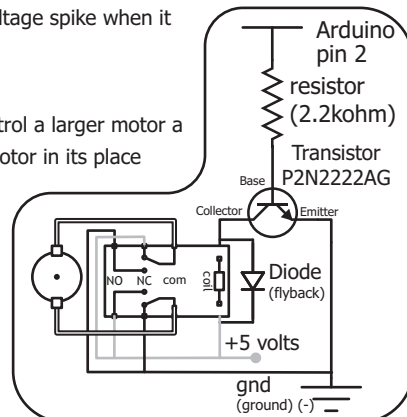
Replace the diode with an LED. You'll see it blink each time it "snubs" the coil voltage spike when it turns off.

Controlling a Motor

In CIRC-03 we controlled a motor using a transistor. However if you want to control a larger motor a relay is a good option. To do this simply remove the red LED, and connect the motor in its place (remember to bypass the 560 Ohm resistor).

Controlling Motor Direction

A bit of a complicated improvement to finish. To control the direction of spin of a DC motor we must be able to reverse the direction of current flow through it. To do this manually we reverse the leads. To do it electrically we require something called an h-bridge. This can be done using a DPDT relay to control the motor's direction, wire up the following circuit. It looks complicated but can be accomplished using only a few extra wires. Give it a try.

**MORE, MORE, MORE:**

More details, where to buy more parts, where to ask more questions:

<http://ardx.org/CIRC11>